EE219: Large-Scale Data Mining: Models and Algorithms
Winter 2018

Project 4: Regression Analysis

Akshya ARUNACHALAM (UID: 904-943-191)
Zhi Ming CHUA (UID: 805-068-401)
Ashwin Kumar KANNAN (UID: 605-035-204)
Vijay RAVI (UID: 805-033-666)

March 5, 2018

Contents

1 Introduction

Regression analysis is a form of predictive modeling technique that estimates the relationship between input and output variables. In this project, basic regression models are used to predict a target variable given a set of potentially relevant variables. A common problem faced in such tasks is over-fitting, which occurs when the model fits too well to the training dataset and fails to capture the general relationship between variables. This problem can be addressed using cross-validation and regularization. Cross-validation checks for over-fitting by using a portion of the training set as a validation set, while regularization penalizes overly complex models, as the improvement in accuracy diminishes with increasing complexity.

2 Dataset

In this project, a network backup dataset is used. It comprises simulated traffic data on a backup system over a network. There are 18,000 data points with the following variables:

- Week number
- Day of the week at which the file back up has started
- Backup start time: Hour of the day
- Workflow ID
- File name
- Backup size: the size of the file that is backed up in that cycle in GB
- Backup time: the duration of the backup procedure in hour

2.1 Backup Sizes

To visualize the dataset, the backup sizes for each workflow type is plotted over a fixed time period.

The day number/index is obtained using Equation (1).

$$i = (w - 1) \times 7 + d \tag{1}$$

where i denotes the day index, w denotes the week number and d denotes the day (Monday to Sunday maps to 1 to 7).

Figure 1 shows the backup sizes for each workflow for the first 20-day period.

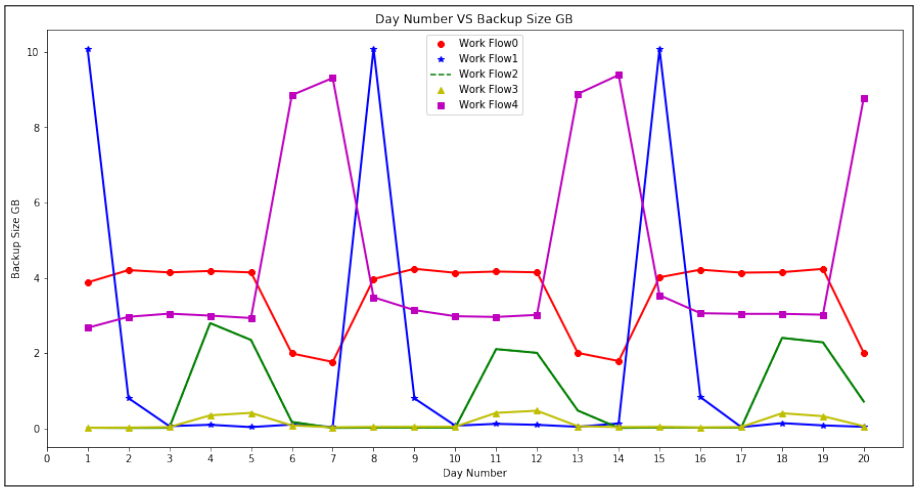


Figure 1: Backup Sizes for 20-Day Period

The same plot for the first 105-day period is shown below.

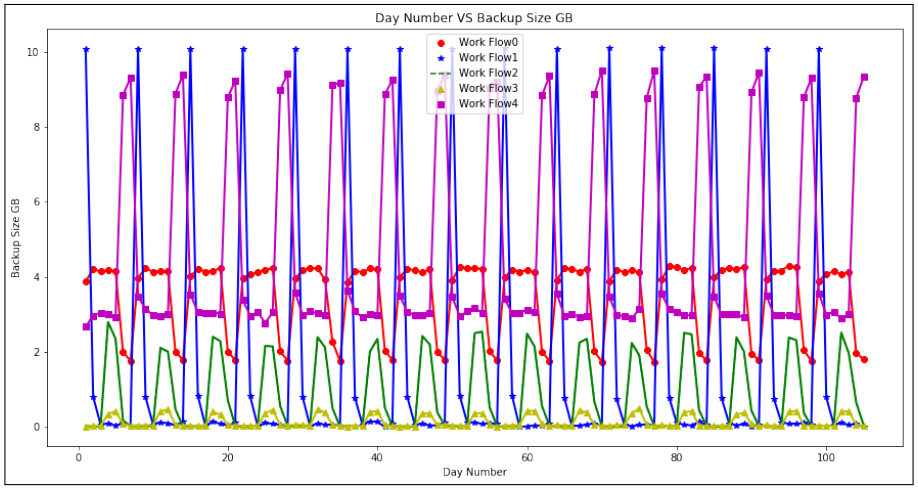


Figure 2: Backup Sizes for 105-Day Period

From both plots, it can be observed that the backup sizes are quasi-periodic with a period of 7 days. For each workflow type, the following patterns are observed:

- **Workflow 0:** Relatively constant at 4 GB on weekdays and drops to around 2 GB during weekends
- **Workflow 1:** Sharp peaks of approximately 10 GB on Mondays
- **Workflow 2:** Virtually no backup takes place except on Thursday and Friday
- **Workflow 3:** Backup size at almost 0 on all days other than Thursday and Friday
- **Workflow 4:** Approximately 3GB on weekdays and increases to around 9GB on weekends

3 Predicting Backup Size

In this section, various regression models are used to predict the backup size of a file given the following variables:

- Week number
- Day of the week
- Hour of the day
- Workflow type
- File name

As the variables used are categorical, two encoding schemes will be investigated, namely the **scalar encoding** and **one-hot encoding**.

To evaluate the performance of each regression model, the training and testing root-mean-square errors (RMSE) from 10-fold cross-validation are computed.

3.1 Linear Regression

The most common linear regression model is the linear least square regression method. It can be formulated as an optimization problem given by Equation (2).

$$\min_{\beta} \quad \|Y - X\beta\|^2 \tag{2}$$

where Y is the vector of target variables, X is the features matrix and β is the coefficient vector. The features matrix includes an all-one column for the bias.

3.1.1 Scalar Encoding

The features are first converted into one-dimensional numerical values using scalar encoding before being applied to the basic linear regression model.

The following RMSE values were obtained.

Training RMSE	0.1036
Testing RMSE	0.1037

Table 1: RMSE with 10-Fold Cross-Validation

Scatter plots of fitted values against true values, and residuals against fitted values are shown in Figures 3 and 4, respectively.

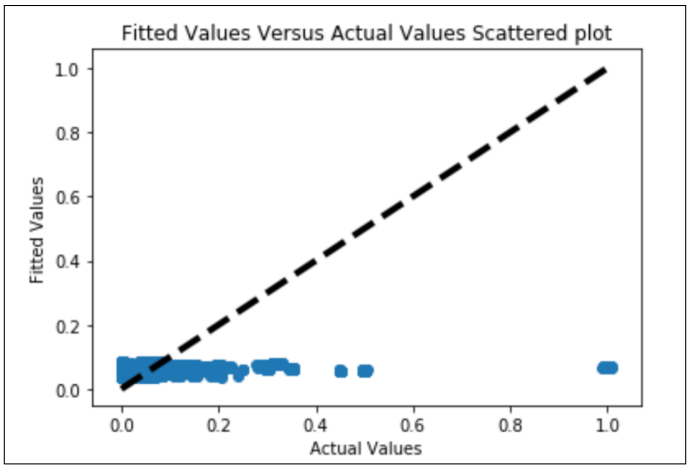


Figure 3: Fitted Values Against Actual Values

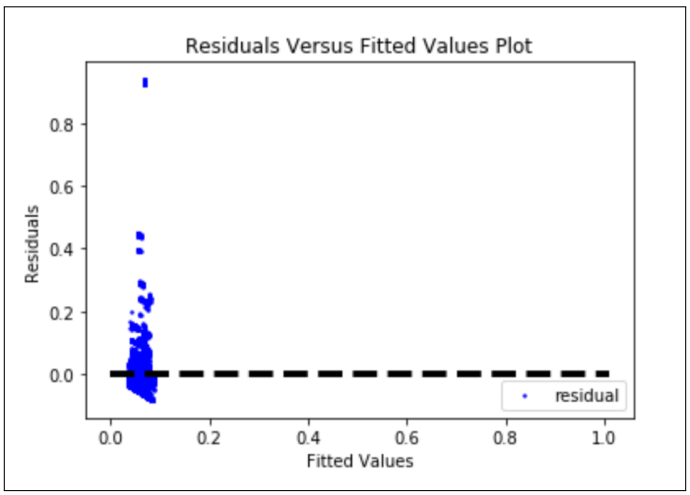


Figure 4: Residuals Against Fitted Values

It can be seen that the model has suboptimal performance when the backup size is large as the predicted values are limited to a narrow range.

The actual and predicted values for every observation are plotted on the same graph in Figure 5.

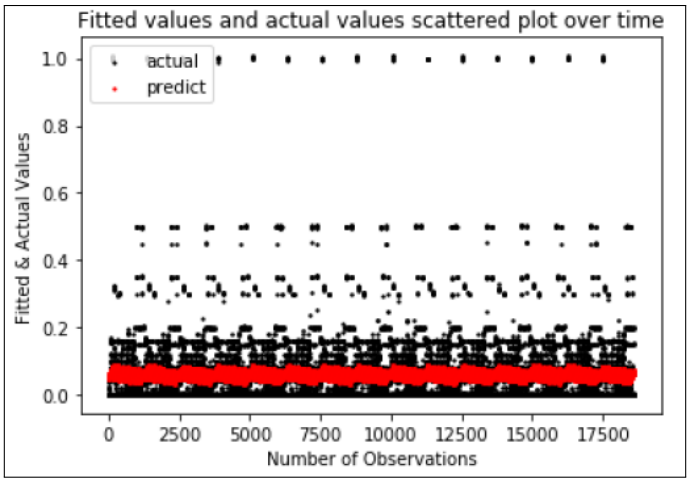


Figure 5: Actual and Predicted Values

3.1.2 Data Preprocessing: Standardization

Standardization centers the mean of each input variable at zero and scales each feature to have unit variance. The effect of standardization when applied to the basic linear regression model is investigated.

The following RMSE values were obtained.

Training RMSE	0.1036
Testing RMSE	0.1037

Table 2: RMSE with 10-Fold Cross-Validation

Scatter plots of fitted values against true values, and residuals against fitted values are shown in Figures 6 and 7, respectively.

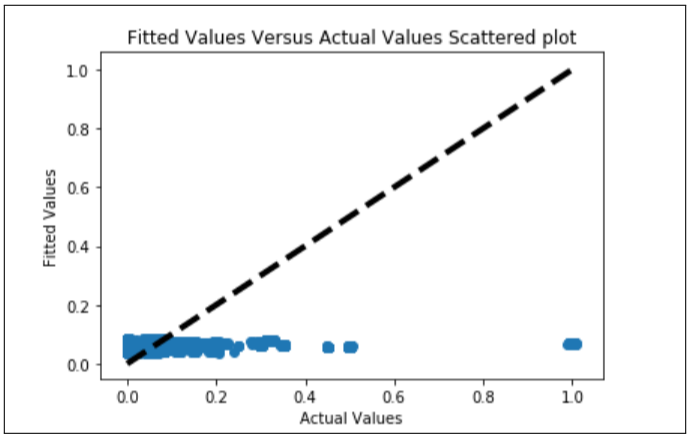


Figure 6: Fitted Values Against Actual Values

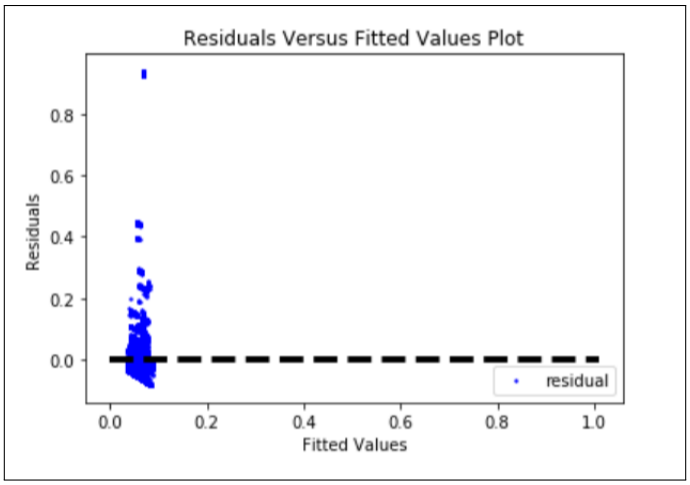


Figure 7: Residuals Against Fitted Values

The results with and without standardization have virtually no difference. This is expected because the centering of the mean simply changes the bias of the linear regression model and the scaling of input variables to have unit variance is equivalent to applying a scaling factor to the respective coefficients. It is, however, important to note that standardizing the target variable will change the performance of the model as the objective function is non-linear.

The actual and predicted values for each observation are plotted below.

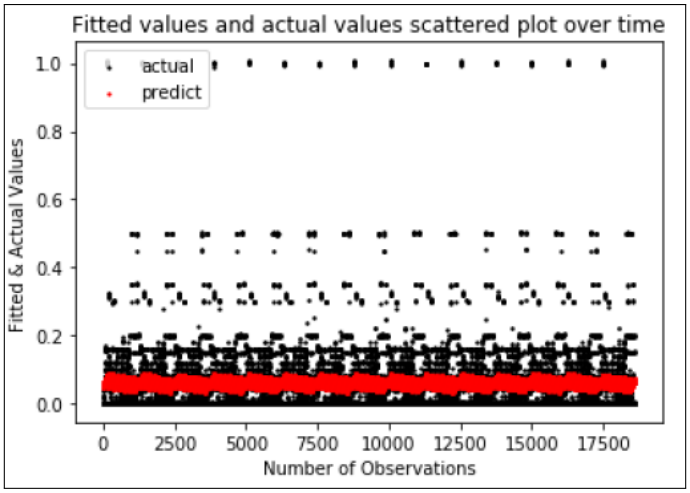


Figure 8: Actual and Predicted Values

3.1.3 Feature Selection

The three most important variables are selected using F-regression and mutual information (MI) regression. They are tabulated in Table 3.

***F*-Regression**	**MI Regression**
Day of the week	Hour of the day
Hour of the day	Workflow type
Workflow type	File name

Table 3: Three Most Important Variables

A linear regression model is trained using the three most important variables from each measure. The respective training and testing RMSE are shown below.

	F-Regression	**MI Regression**
Training RMSE	0.1036	0.1037
Testing RMSE	0.1037	0.1038

Table 4: RMSE Using Three Most Important Variables

The improvement over the initial model with variables selected using F-regression is minimal and the RMSE increased slightly for variables selected using MI regression.

Plots for fitted values against true values, and residuals against fitted values, with variables selected using F-regression, are shown below.

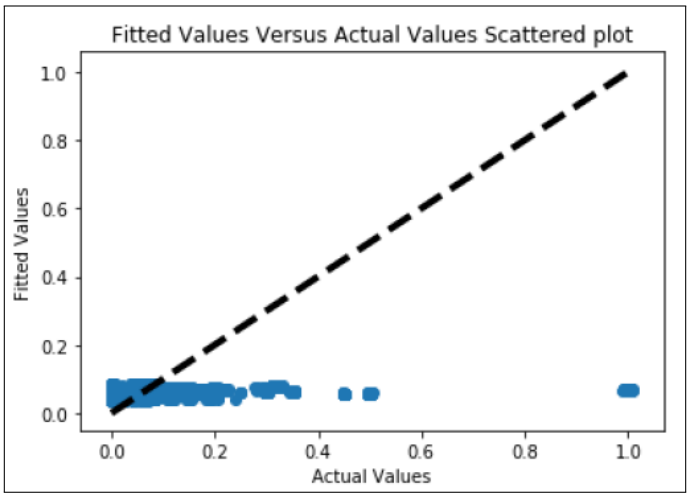


Figure 9: Fitted Values Against Actual Values

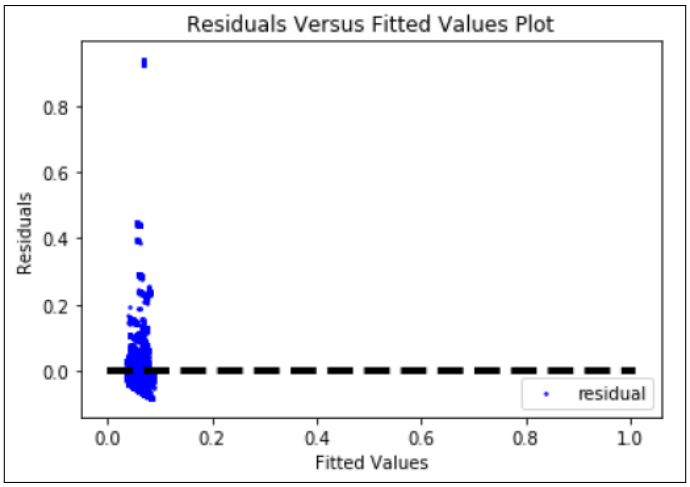


Figure 10: Residuals Against Fitted Values

The actual and predicted values for all observations are plotted on the same graph below.

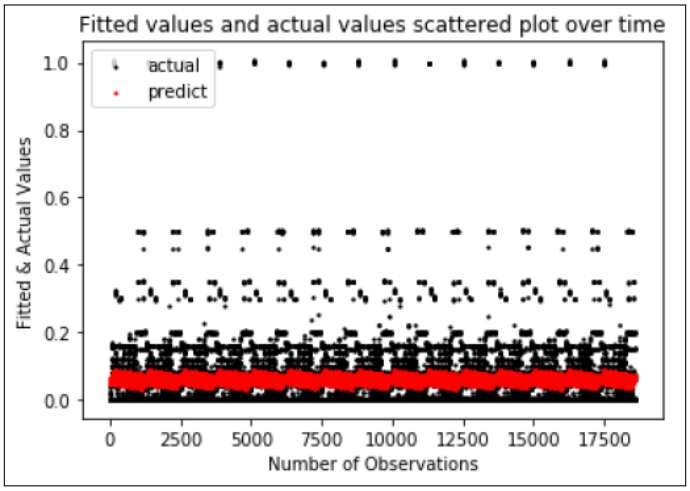
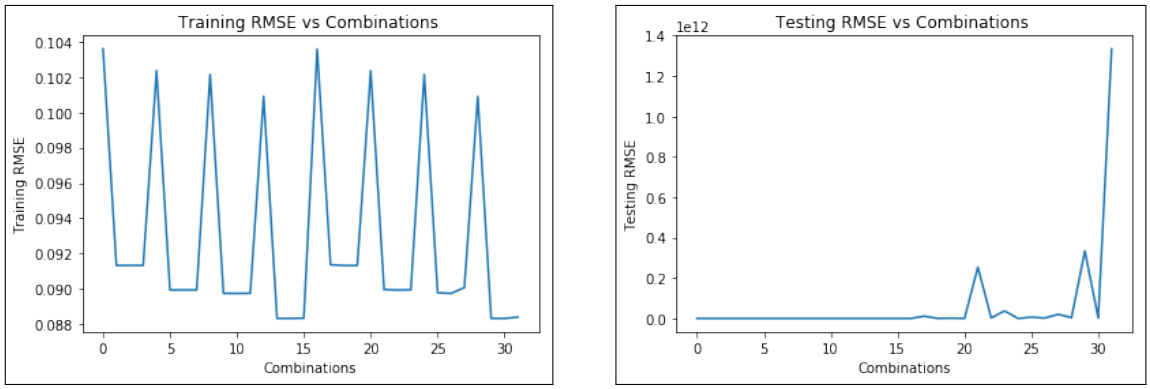


Figure 11: Actual and Predicted Values

3.1.4 Feature Encoding

For each of the categorical variables, there are 2 encoding schemes that can be employed. Hence, there are $2^5 = 32$ possible combinations of encoding. In this section, the performance for each combination is compared and the best combination is reported.

The average RMSE for each combination is plotted in Figure 12.



(a) Training (b) Testing

Figure 12: Average RMSE

The combinations that yielded the best performance in terms of training and testing RMSE are tabulated in Table 5.

	Best Training RMSE	**Best Testing RMSE**
Week Number	Scalar	Scalar
Day of the Week	One-Hot	One-Hot
Hour of the Day	One-Hot	One-Hot
Workflow Type	Scalar	One-Hot
File Name	One-Hot	Scalar
Training RMSE	**0.088336**	0.088337
Testing RMSE	0.088505	**0.088504**

Table 5: Best Encoding Combinations

For both training and testing, week number is best encoded using scalar encoding. This is probably due to the weekly-periodic nature of the backup sizes as the "similarity" between weeks are approximately equal.

Day of the week and hour of the day are best encoded using one-hot encoding. This may be due to backup sizes between days, as well as between hours, being highly independent or uncorrelated.

For workflow type and file name, there is probably a small trade-off between the number of features and the testing fit. As there are 30 file names and 5 workflow types, when the file names are one-hot encoded, there are more features and hence the model will fit the training data slightly better. On the other hand, when the workflow types are one-hot encoded, the number of features is fewer and hence fits the testing data better.

Overall, the performance for both encoding combinations are comparable, and they both out-performed the model trained with all-scalar-encoded inputs. To visualize the improvement, scatter plots of fitted values against true values, and residuals against fitted values are shown in Figures 13 and 14, respectively.

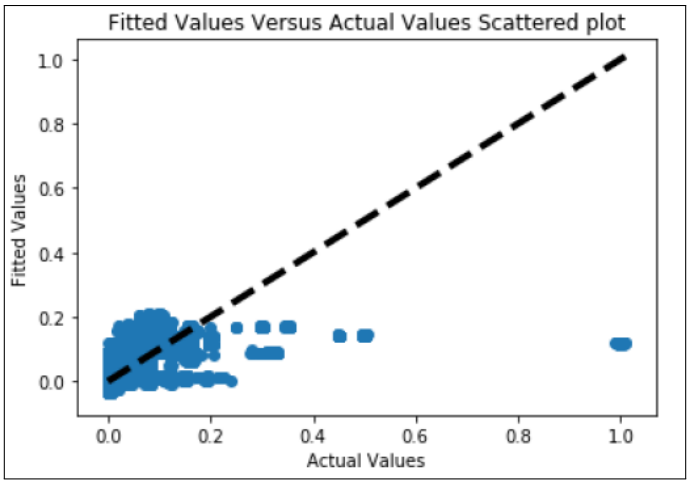


Figure 13: Fitted Values Against Actual Values

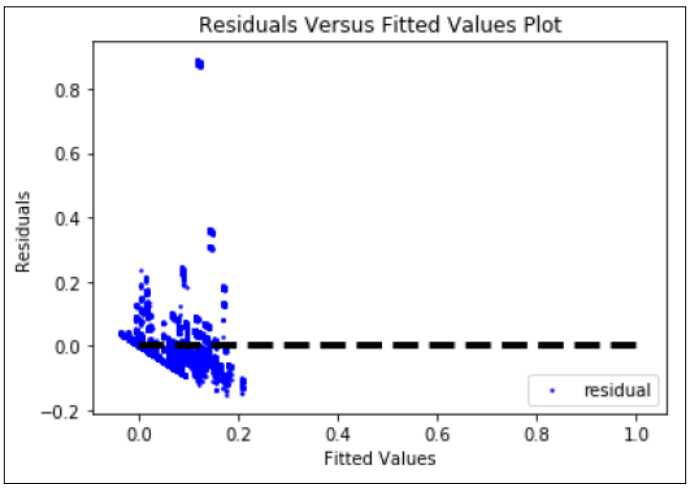


Figure 14: Residuals Against Fitted Values

The predicted value range widened and hence a lower RMSE is obtained. However, the performance on predicting large backup sizes is still poor.

The actual and predicted values are plotted on the same graph in Figure 15.

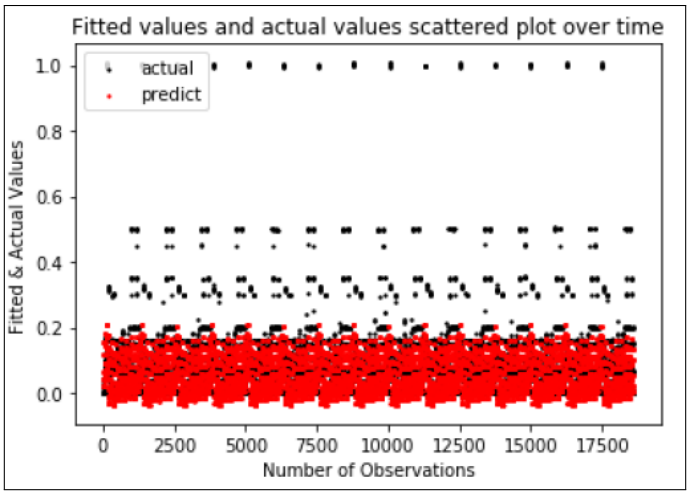


Figure 15: Actual and Predicted Values

3.1.5 Regularization

From Figure 12, it can be seen that there are obvious increases in testing RMSE compared to training RMSE for some combinations of encoding schemes. This is because the model fits the training data too well that the general trend is not captured, resulting in poor performance for unseen datasets (testing data). It can be thought of as the model fitting to the variance of the training data instead of just the mean. For these models, the magnitude of the coefficients are significantly larger.

To address this problem, regularization is introduced to the loss function. It aims to penalize large coefficients and can help prevent over-fitting.

Three regularizers are investigated in this section.

1. Ridge: $\min_{\beta} \|Y - X\beta\|^2 + \alpha\|\beta\|_2^2$
2. Lasso: $\min_{\beta} \|Y - X\beta\|^2 + \alpha\|\beta\|_1$
3. Elastic Net: $\min_{\beta} \|Y - X\beta\|^2 + \lambda_1\|\beta\|_1 + \lambda_2\|\beta\|_2^2$

For each regularizer, the model is optimized over the 32 possible encoding combinations and hyper-parameters in factors of 10. It is observed that the RMSE initially decreases then increases with increasing hyper-parameter values. This is because increasing penalty reduces the error due to variance but introducing too large a penalty prevents the model from fitting the general trend, resulting in large errors as the bias is not sufficiently captured.

Compared to the best non-regularized model, the best regularized models have significantly smaller coefficient magnitudes, with difference in orders of magnitude of at least -10.

The testing RMSE for each regularization implemented is tabulated below.

	Ridge	**Lasso**	**Elastic Net**
Testing RMSE	0.08850	0.08851	0.08851

Table 6: Testing RMSE for Different Regularizations

The ridge regularizer produced the lowest testing RMSE and the scatter plots of fitted values against true values, and residuals against fitted values, using this regularizer, are shown in Figures 16 and 17, respectively.

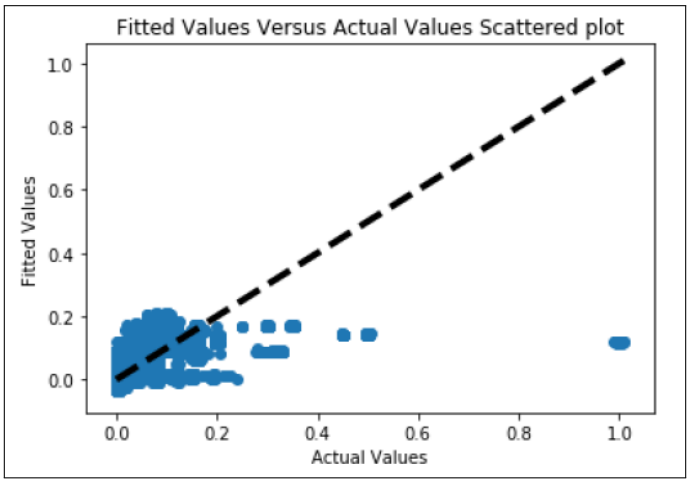


Figure 16: Fitted Values Against Actual Values

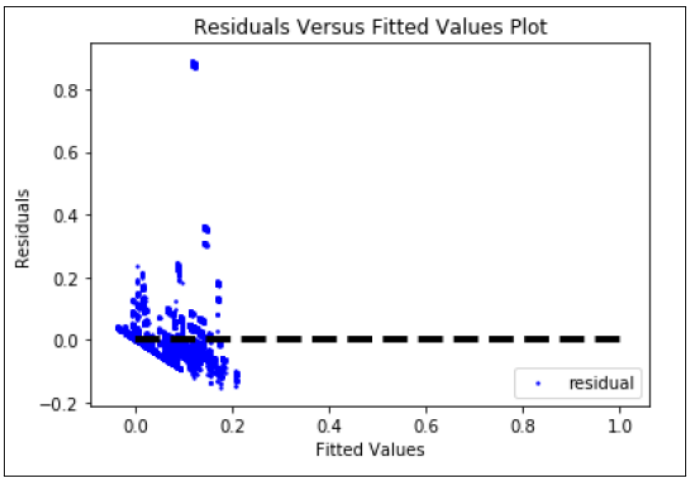


Figure 17: Residuals Against Fitted Values

The actual and predicted values for every observation are plotted on the same graph in Figure 18.

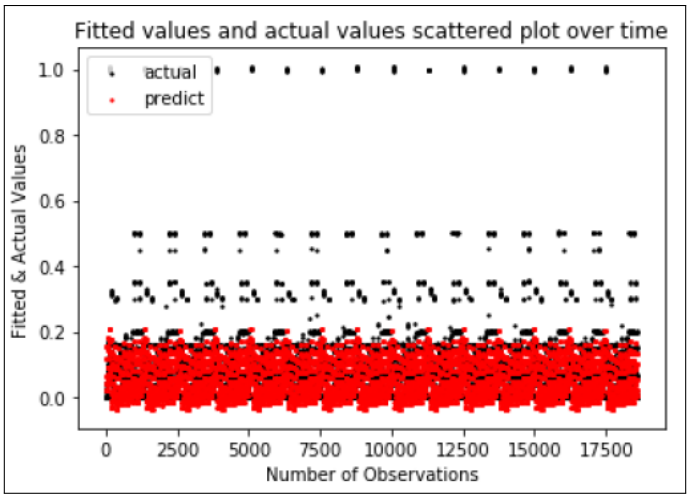


Figure 18: Actual and Predicted Values

3.2 Random Forest Regression Model

In this section, random forest regression is used to predict the backup sizes of files.

3.2.1 Initial Model

The parameters of the model are set to the following initial values:

- Number of trees: 20
- Depth of each tree: 4
- Bootstrap: True
- Maximum number of features: 5

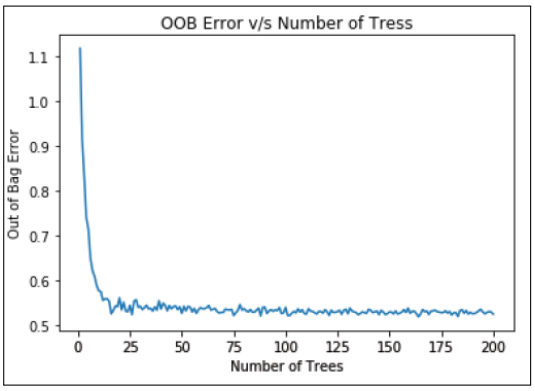
The training RMSE, testing RMSE and out-of-bag error are tabulated in Table 7

Training RMSE	0.06050
Testing RMSE	0.06062
Out-of-Bag Error	0.3393

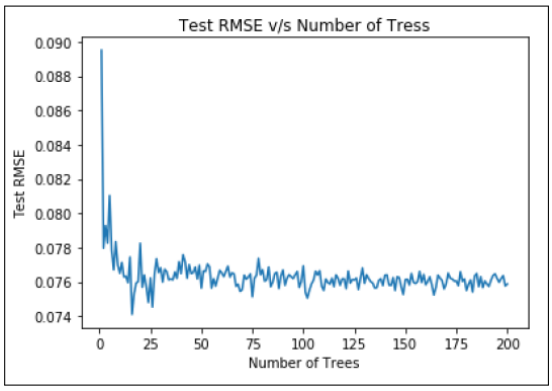
Table 7: Initial Model Errors

3.2.2 Varying Number of Trees and Maximum Number of Features

The maximum number of features is swept from 1 to 5 and for each of the value taken, the number of trees is swept from 1 to 200. For the different maximum number of features, the out-of-bag error and testing RMSE are plotted against the number of trees.

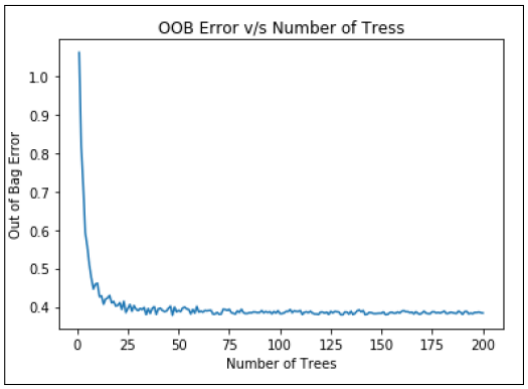


(a) Out-of-Bag Error

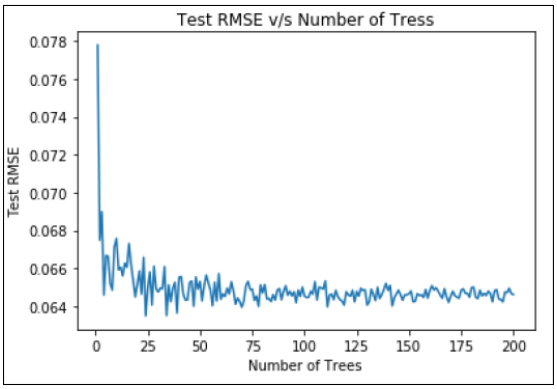


(b) Testing RMSE

Figure 19: Maximum Number of Features = 1

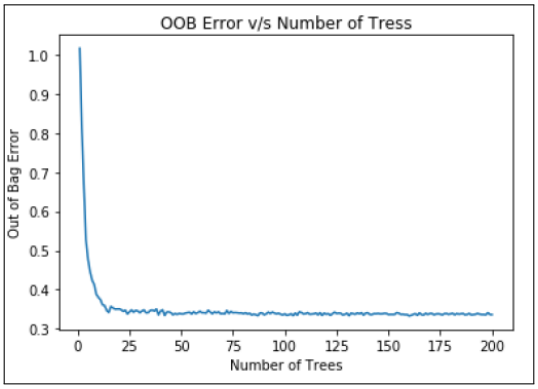


(a) Out-of-Bag Error

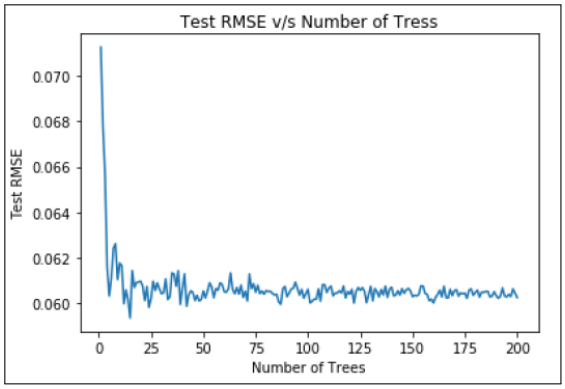


(b) Testing RMSE

Figure 20: Maximum Number of Features = 2

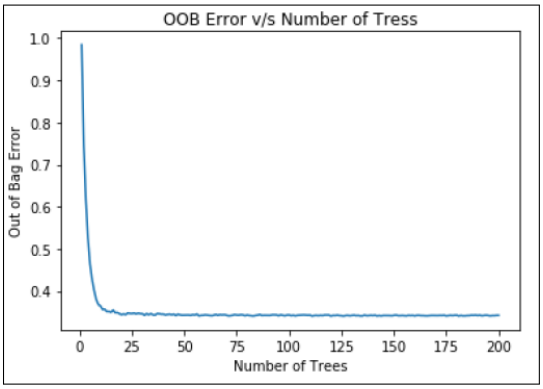


(a) Out-of-Bag Error

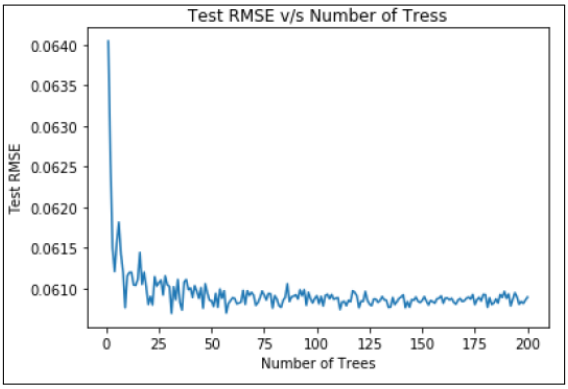


(b) Testing RMSE

Figure 21: Maximum Number of Features = 3

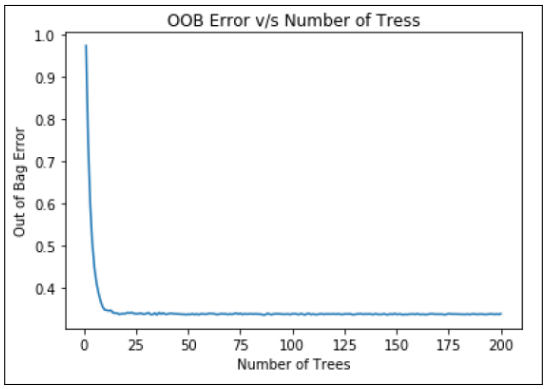


(a) Out-of-Bag Error

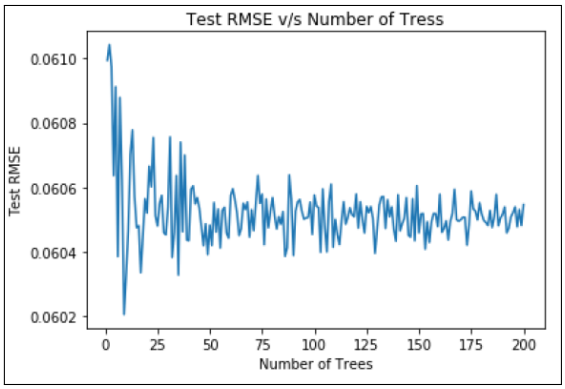


(b) Testing RMSE

Figure 22: Maximum Number of Features = 4



(a) Out-of-Bag Error



(b) Testing RMSE

Figure 23: Maximum Number of Features = 5

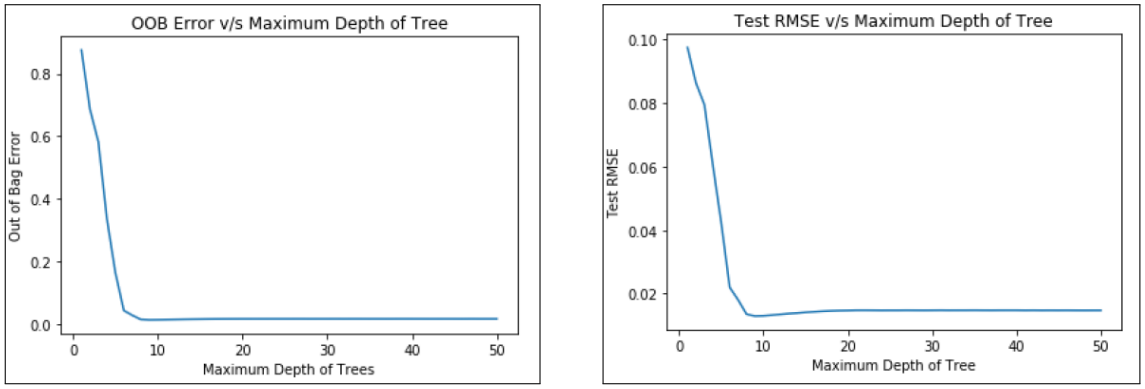
The lowest testing RMSE and out-of-bag error are achieved when the number of trees is set to 25 and the maximum number of features is 5. The errors produced by the model with these parameters are tabulated below.

Training RMSE	0.06033
Testing RMSE	0.06048
Out-of-Bag Error	0.3380

Table 8: Best Model Errors

3.2.3 Varying Maximum Depth of Tree

Using the parameters determined in the previous section, the maximum depth of the trees is swept from 1 to 50 to obtain the maximum tree depth that yields the best performance. The out-of-bag error and testing RMSE are plotted against the the maximum depth in Figure 24.



(a) Out-of-Bag Error

(b) Testing RMSE

Figure 24: Error Values for Varying Maximum Depth of Tree

The best performance is achieved when the maximum depth is set at 9. The error values at this maximum depth are tabulated below.

Training RMSE	0.01153
Testing RMSE	0.01293
Out-of-Bag Error	0.01497

Table 9: Best Model Errors

Therefore, to achieve the best performance, the model parameters are selected as follows:

- Number of trees: 25
- Depth of each tree: 9
- Bootstrap: True
- Maximum number of features: 5

The scatter plots of fitted values against true values, and residuals against fitted values are shown in Figures 25 and 26, respectively.

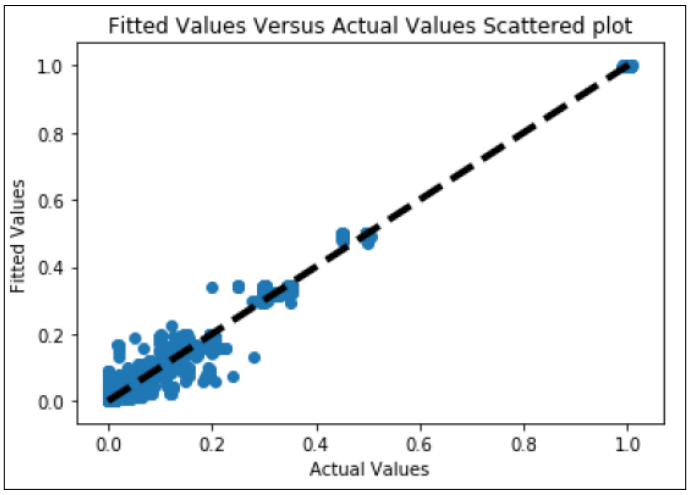


Figure 25: Fitted Values Against Actual Values

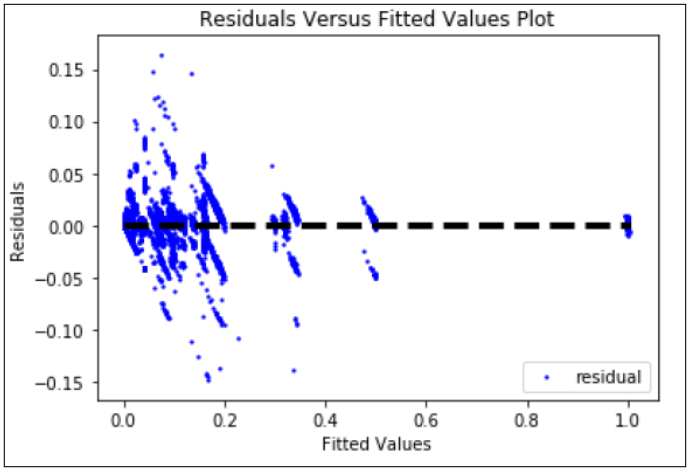


Figure 26: Residuals Against Fitted Values

The actual and predicted values for each data point are plotted on the same graph in Figure 27.

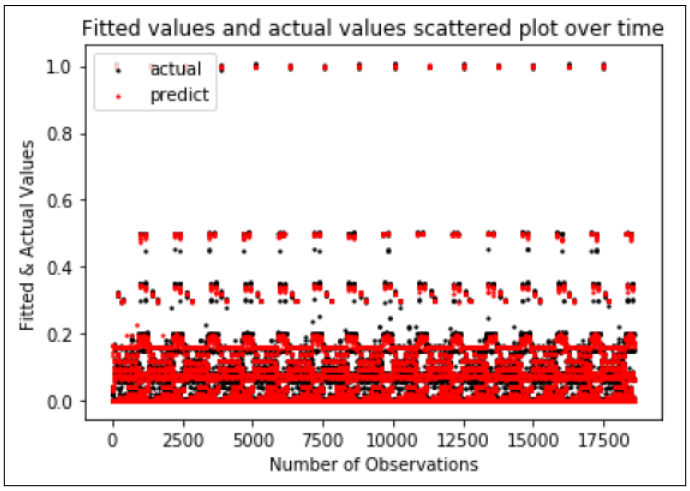


Figure 27: Actual and Predicted Values

3.2.4 Feature Importance

The feature importances were obtained using the `RandomForestRegressor` function and the values for trees with maximum depth 9 and 4 are shown in Tables 10 and 11, respectively.

	Feature Importance
Week Number	0.001723
Day of the Week	0.1968
Hour of the Day	0.3985
Workflow Type	0.1435
File Name	0.2594

Table 10: Feature Importance: Max Depth = 9

	Feature Importance
Week Number	3.259×10^{-6}
Day of the Week	2.776×10^{-1}
Hour of the Day	1.551×10^{-1}
Workflow Type	1.615×10^{-1}
File Name	4.057×10^{-1}

Table 11: Feature Importance: Max Depth = 4

3.2.5 Visualization

To visualize the decision trees, a tree is selected from the best random forest with max depth = 4 and its structure is displayed in Figure 28, which is obtained using the `scikit` functionality `exportGraphViz`. The depth is clipped at four although the best decision tree is of depth 9 because it is difficult to visualize trees with depth greater than four.

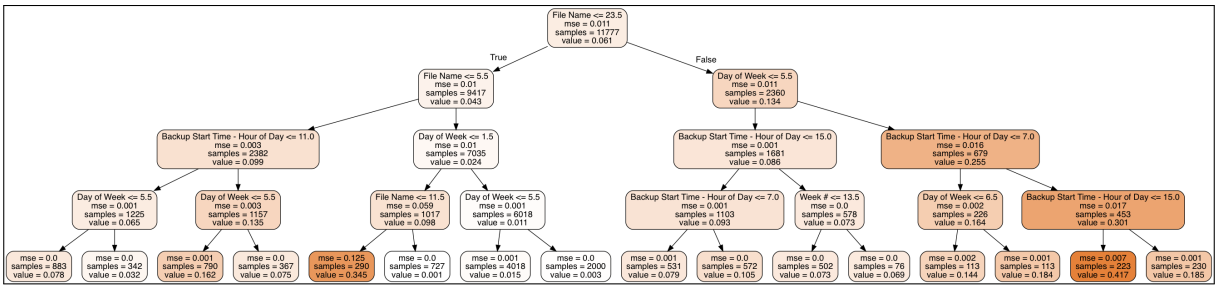


Figure 28: Tree in Best Random Forest (Max Depth = 4)

The root node in the decision tree is of the feature 'File Name', which corresponds to the most important feature according to the regressor, which can be found in Table 11.

3.3 Neural Network Regression Model

In this section, a neural network with a single hidden layer is used to predict the backup sizes. The features are one-hot encoded before being fed to the input layer of the neural network.

The number of hidden neurons and the activation function (relu, logistic, tanh) are varied to determine the optimal combination. The performance is evaluated using the testing RMSE, which is plotted as a function of the number of hidden neurons for different activation functions below.

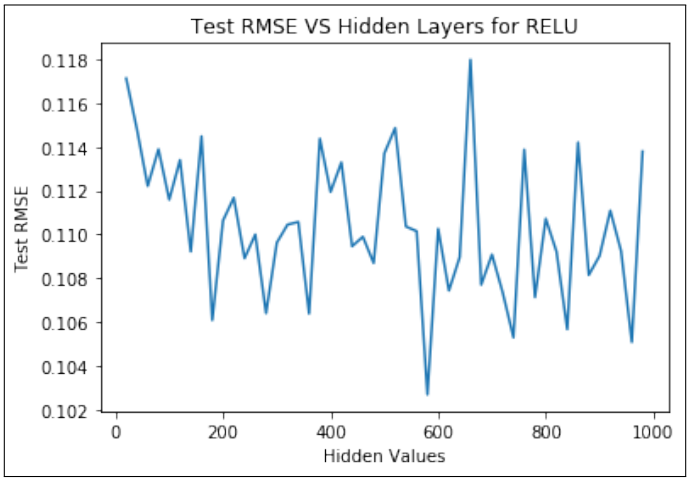


Figure 29: Testing RMSE Against Number of Hidden Neurons (relu)

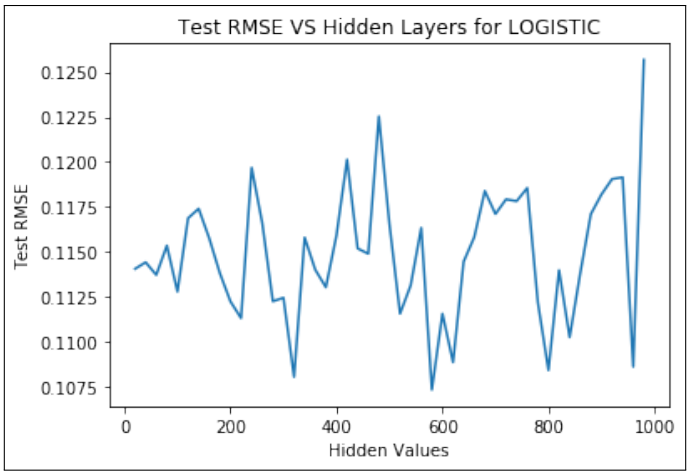


Figure 30: Testing RMSE Against Number of Hidden Neurons (logistic)

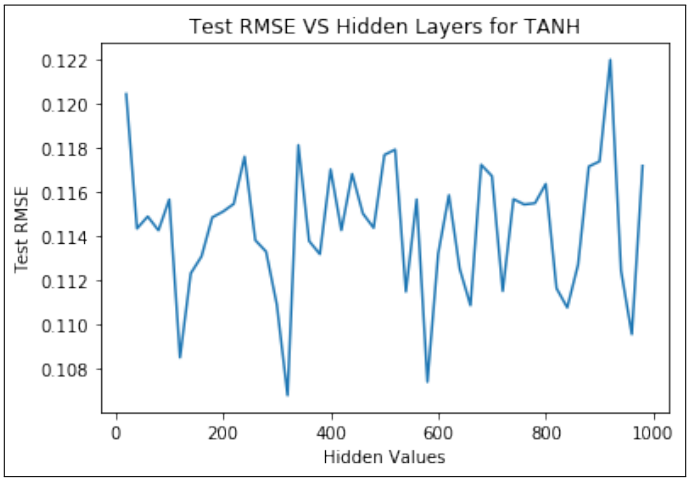


Figure 31: Testing RMSE Against Number of Hidden Neurons (tanh)

The number of hidden neurons that produced the best neural network regression model for each activation function is tabulated in Table 12 with their respective testing RMSE.

	relu	**logistic**	**tanh**
Number of Hidden Neurons	580	580	320
Testing RMSE	0.1027	0.1073	0.1068

Table 12: Testing RMSE for Varying Number of Hidden Neurons and Activation Function

Neural network regression with **580 hidden neurons** and **relu activation function** produced the lowest RMSE. The scatter plots of fitted values against true values, and residuals against fitted values are shown in Figures 32 and 33, respectively.

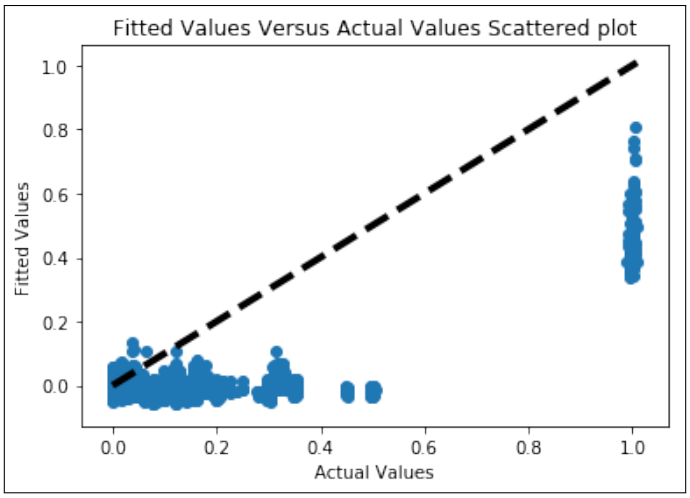


Figure 32: Fitted Values Against Actual Values

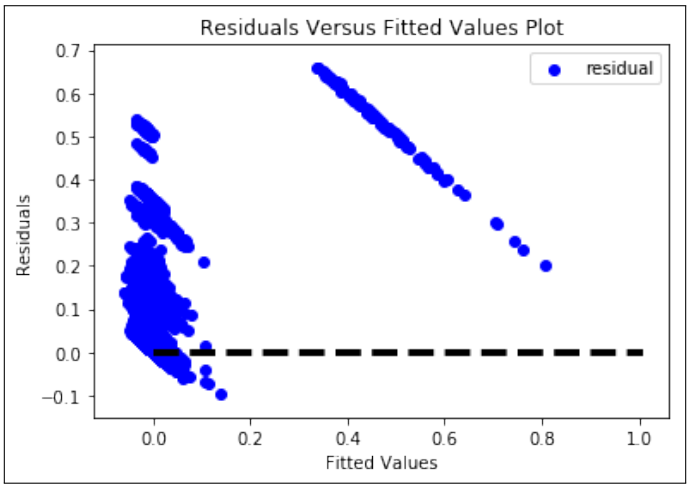


Figure 33: Residuals Against Fitted Values

The actual and predicted values for every observation are plotted on the same graph in Figure 34.

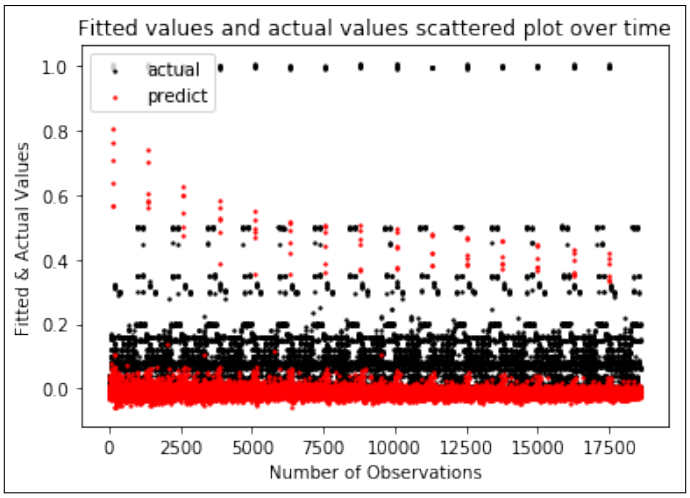


Figure 34: Actual and Predicted Values

3.4 Predicting Backup Sizes For Each Workflow Type

3.4.1 Linear Regression

In this section, a separate linear regression model with 10-fold cross-validation is used to estimate backup sizes for each workflow type. The training and test RMSE values for each workflow type are reported in the table below.

Workflow Type	Training RMSE	Testing RMSE
0	0.03584	0.03589
1	0.1488	0.1489
2	0.04291	0.04291
3	0.007244	0.007244
4	0.08592	0.08600

Table 13: Training and Testing RMSE For Each Workflow Type

The scatter plots of fitted values against true values, residuals against fitted values, and actual and predicted values for each data point are plotted. This is repeated for each of the workflow types.

3.4.1.1 Workflow Type 0

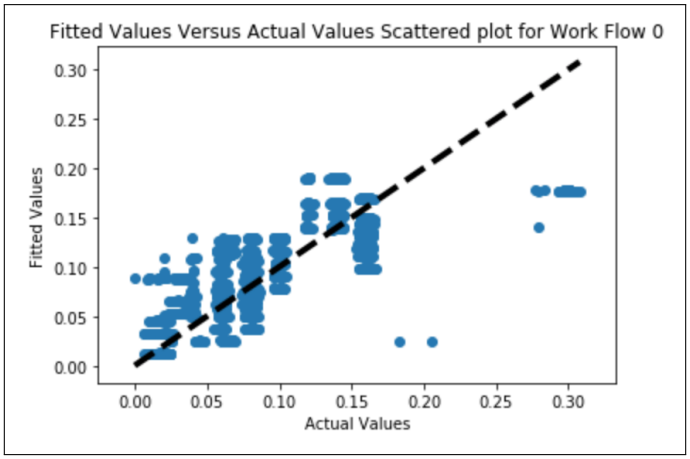


Figure 35: Fitted Values Against Actual Values

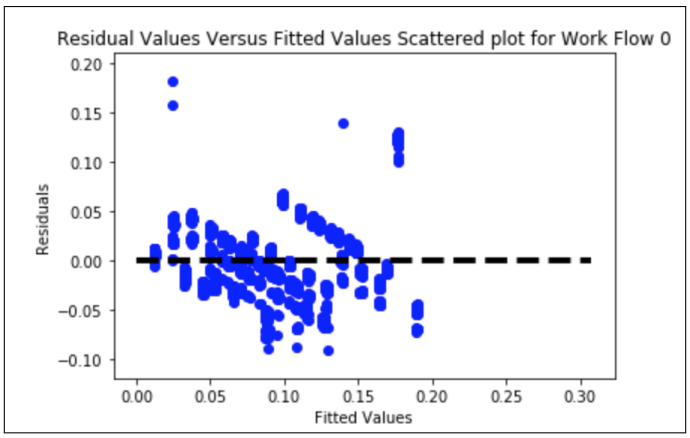


Figure 36: Residuals Against Fitted Values

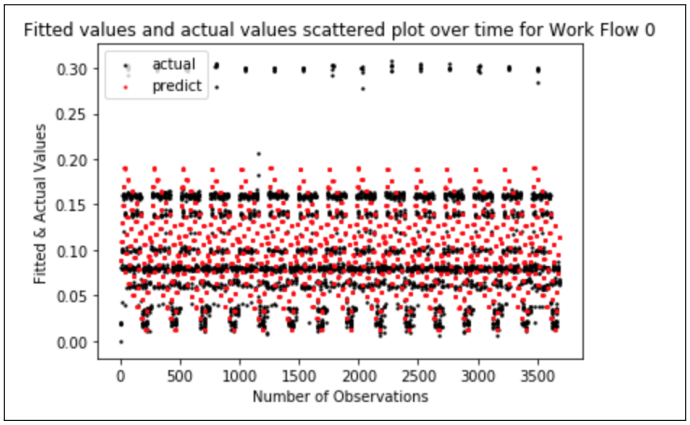


Figure 37: Actual and Predicted Values

3.4.1.2 Workflow Type 1

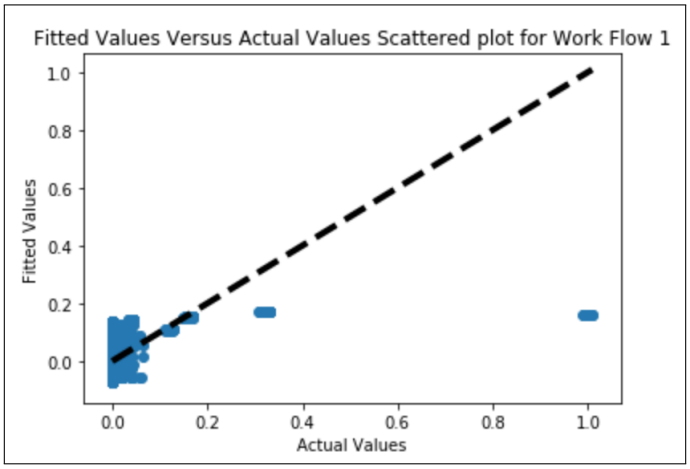


Figure 38: Fitted Values Against Actual Values

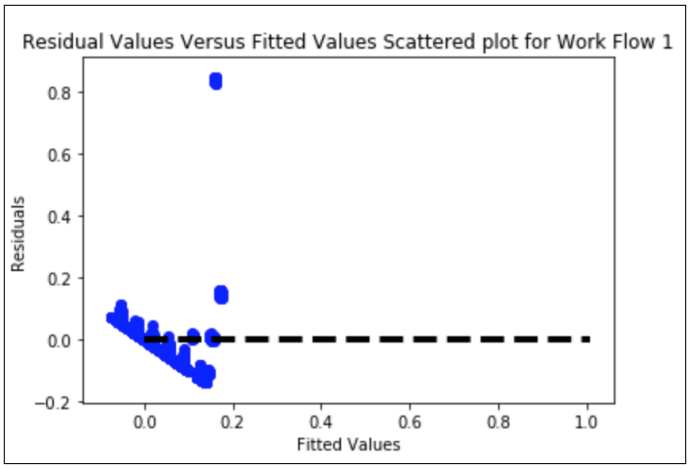


Figure 39: Residuals Against Fitted Values

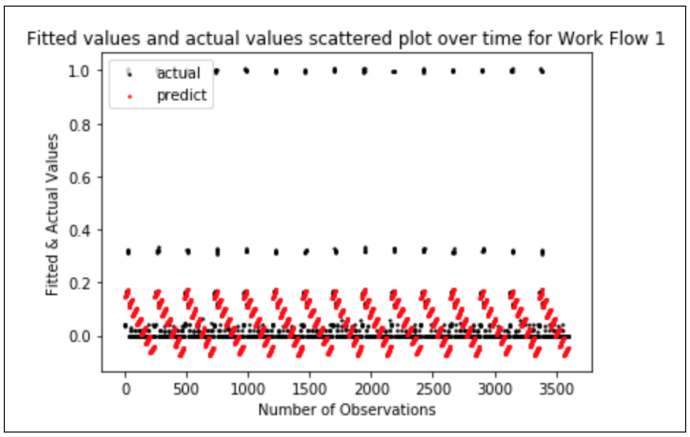


Figure 40: Actual and Predicted Values

3.4.1.3 Workflow Type 2

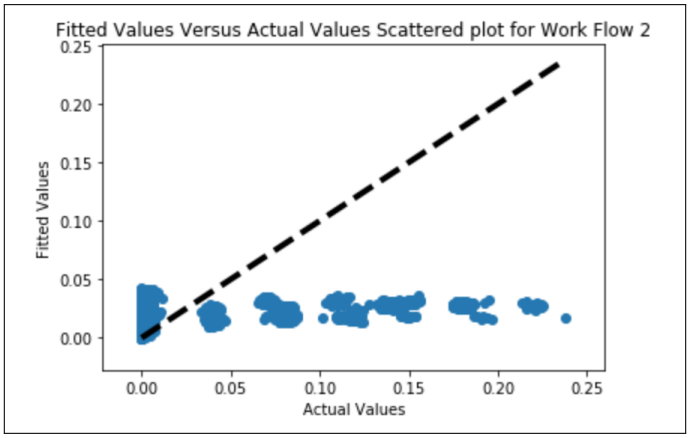


Figure 41: Fitted Values Against Actual Values

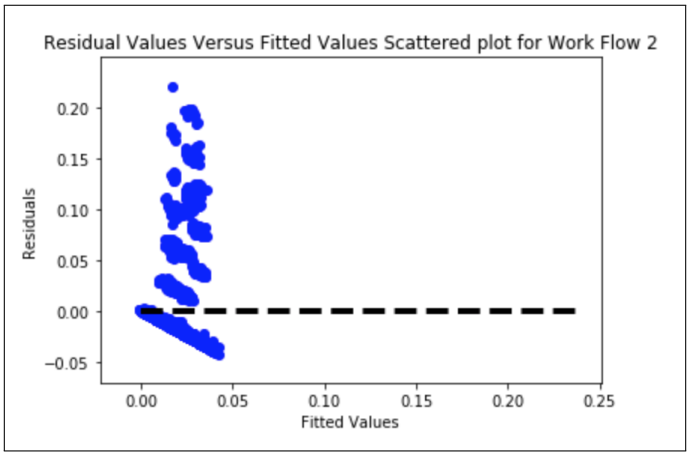


Figure 42: Residuals Against Fitted Values

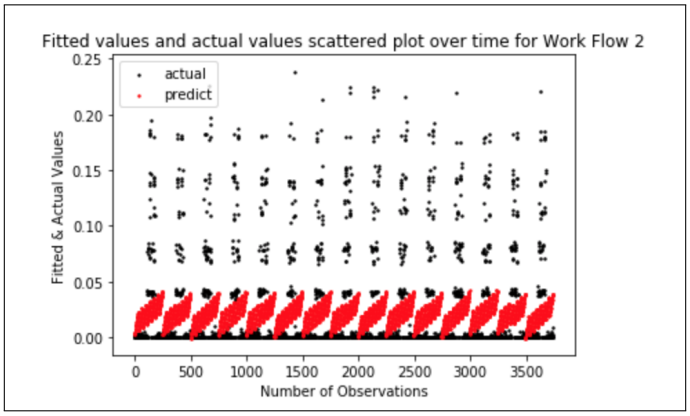


Figure 43: Actual and Predicted Values

3.4.1.4 Workflow Type 3

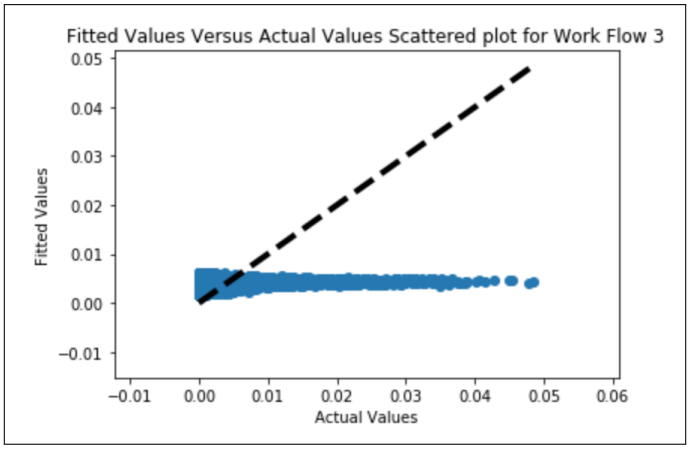


Figure 44: Fitted Values Against Actual Values

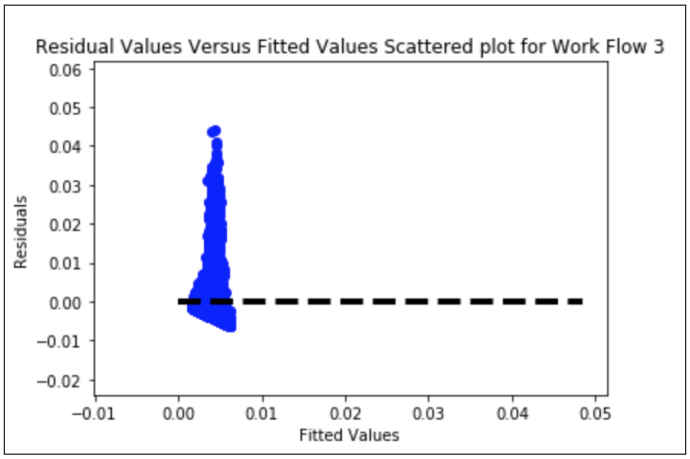


Figure 45: Residuals Against Fitted Values

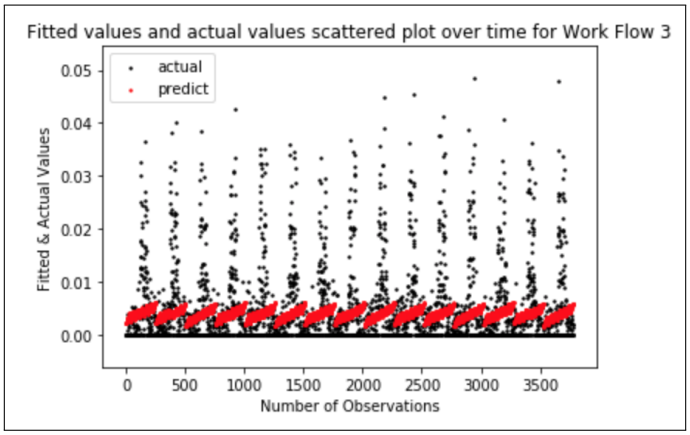


Figure 46: Actual and Predicted Values

3.4.1.5 Workflow Type 4

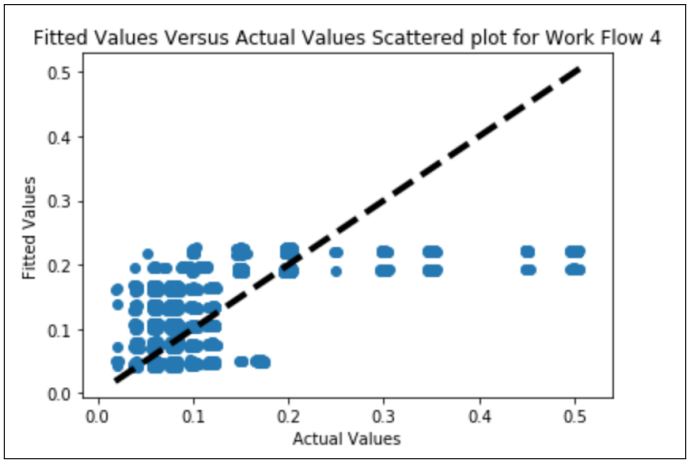


Figure 47: Fitted Values Against Actual Values

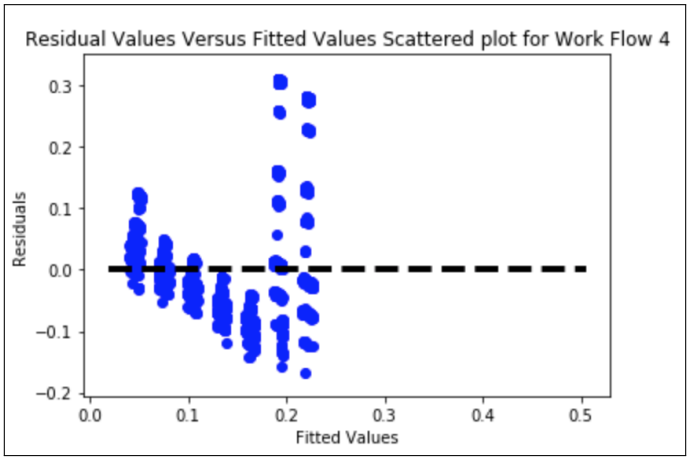


Figure 48: Residuals Against Fitted Values

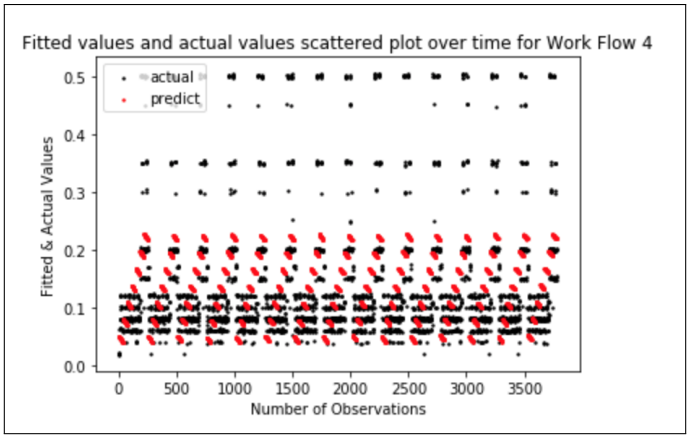


Figure 49: Actual and Predicted Values

3.4.1.6 Analysis

Compared to when estimating backup sizes for all workflow types together, predicting for each workflow type separately yields a better performance for all workflow types except type 1. This is expected because each workflow type has a different pattern and hence the models that fit each of them best are sufficiently distinct. The poor performance for workflow type 1 could be attributed to a sharp peak on a single day, leading to high variance.

3.4.2 Polynomial Regression

In this section, polynomial functions of variables are applied as input to the linear regression model. The effect of increasing the degree of the polynomial on the prediction performance is investigated.

The threshold on the degree of the fitted polynomial beyond which the generalization error worsens can be determined by observing the training RMSE and testing RMSE curves. Generally, the threshold is at where the training RMSE continues decreasing while the testing RMSE starts increasing.

3.4.2.1 Workflow Type 0

The training RMSE and testing RMSE are plotted against the degree of the polynomial used in Figure 50.

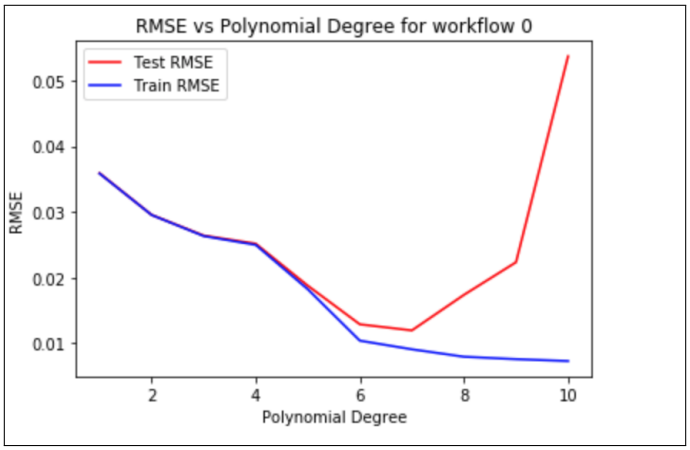


Figure 50: Training and Testing RMSE Against Degree of Polynomial

The threshold on the degree of the polynomial is 7 and the scatter plots shown are for the prediction model set at this parameter. The testing RMSE obtained is 0.01194.

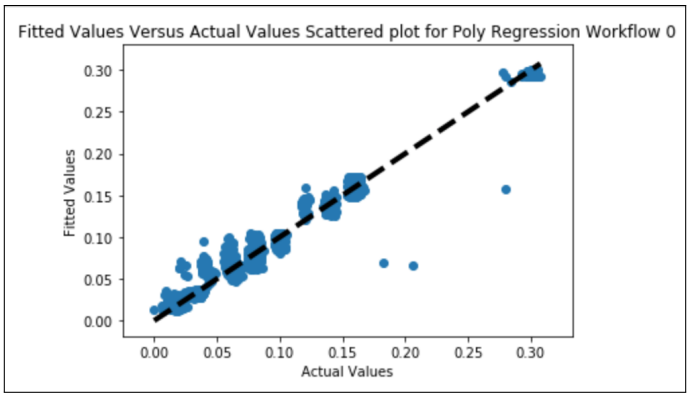


Figure 51: Fitted Values Against Actual Values

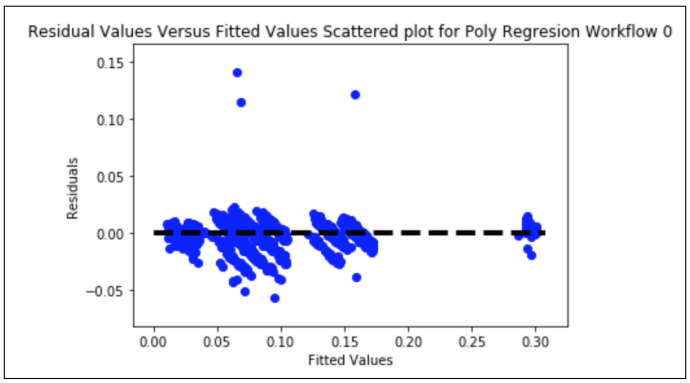


Figure 52: Residuals Against Fitted Values

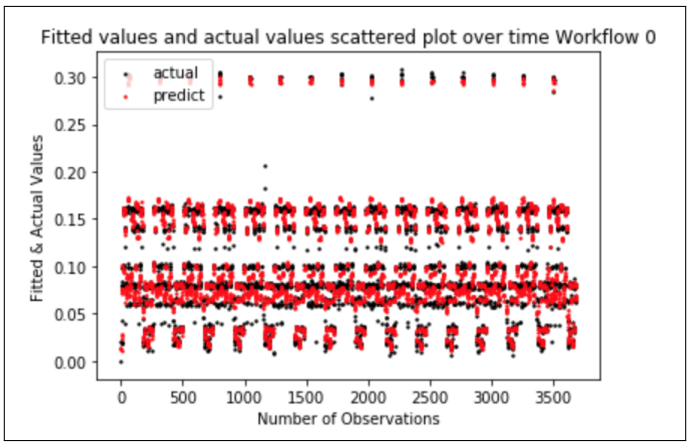


Figure 53: Actual and Predicted Values

3.4.2.2 Workflow Type 1

The training RMSE and testing RMSE are plotted against the degree of the polynomial used.

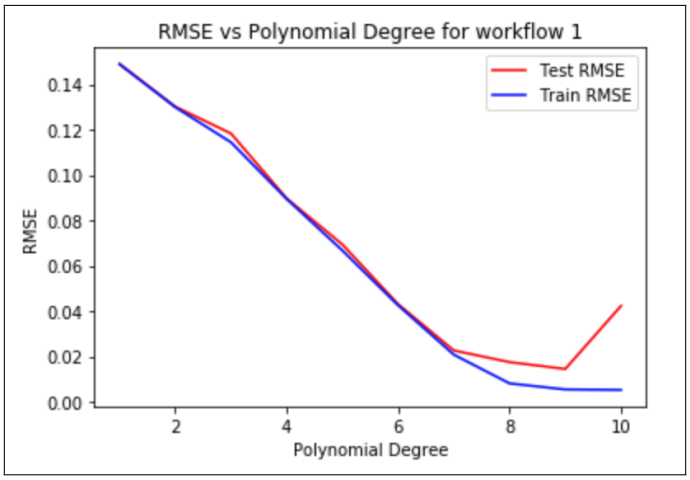


Figure 54: Training and Testing RMSE Against Degree of Polynomial

The scatter plots shown are for the prediction model where the degree of the polynomial is set at the threshold, which is 9. The testing RMSE obtained is 0.01445.

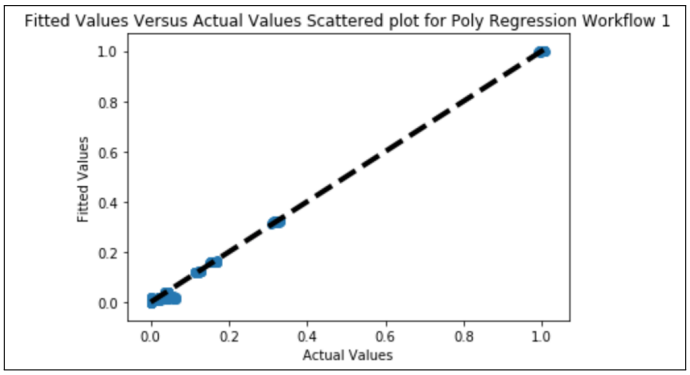


Figure 55: Fitted Values Against Actual Values (Type 1)

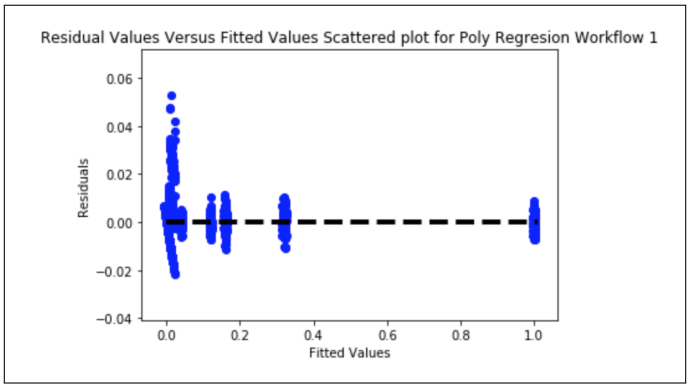


Figure 56: Residuals Against Fitted Values (Type 1)

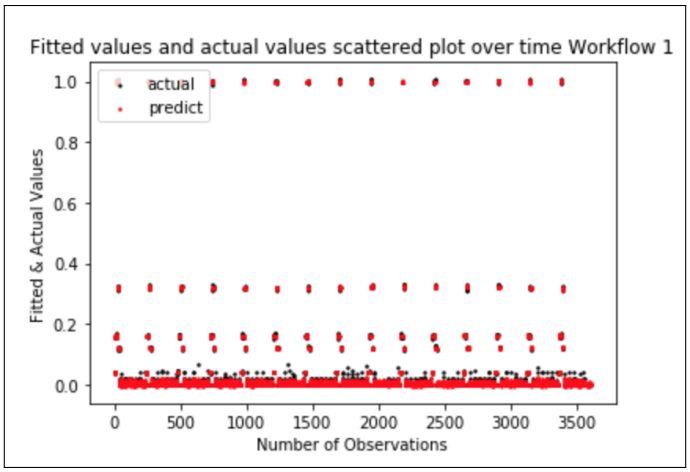


Figure 57: Actual and Predicted Values (Type 1)

3.4.2.3 Workflow Type 2

The training RMSE and testing RMSE are plotted against the degree of the polynomial used.

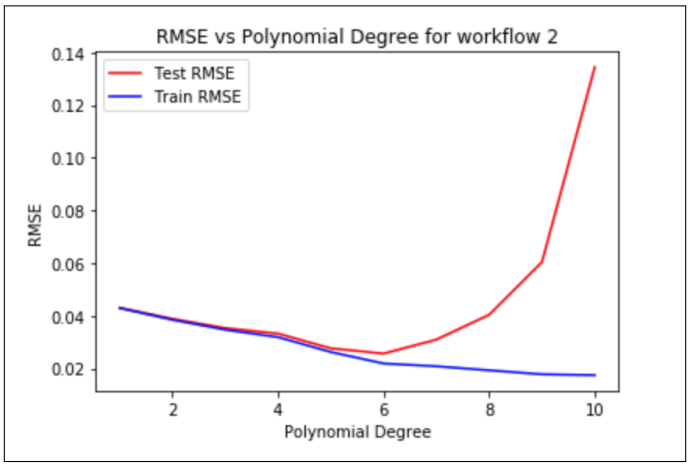


Figure 58: Training and Testing RMSE Against Degree of Polynomial

The threshold on the degree of the polynomial is 6 and the scatter plots shown are for the prediction model set at this parameter. The testing RMSE obtained is 0.02565.

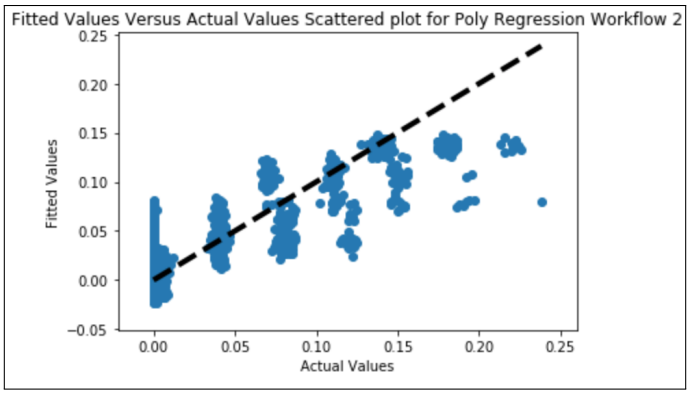


Figure 59: Fitted Values Against Actual Values (Type 2)

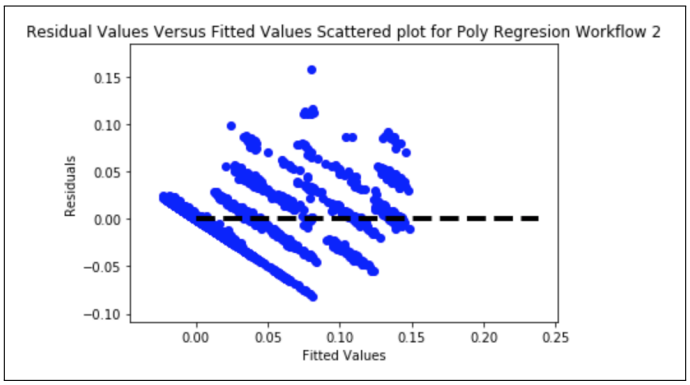


Figure 60: Residuals Against Fitted Values (Type 2)

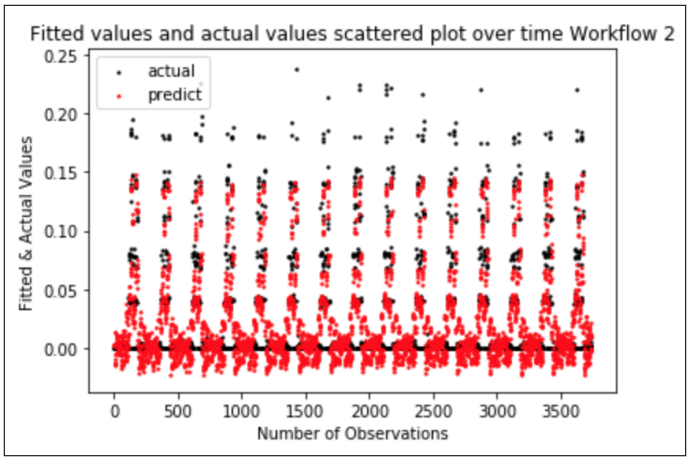


Figure 61: Actual and Predicted Values (Type 2)

3.4.2.4 Workflow Type 3

The training RMSE and testing RMSE are plotted against the degree of the polynomial used below.

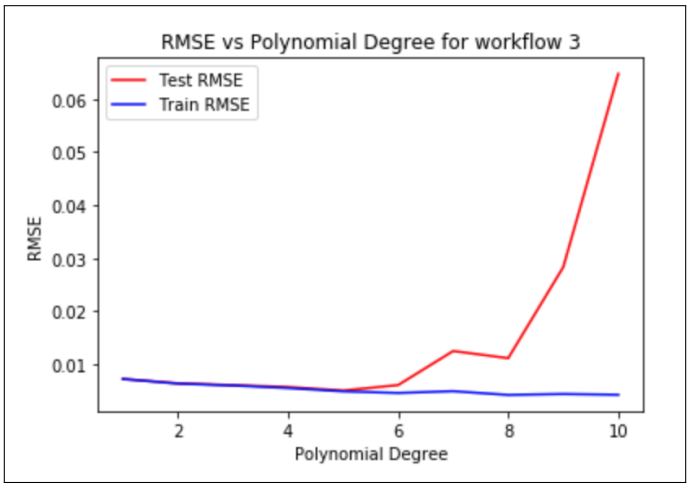


Figure 62: Training and Testing RMSE Against Degree of Polynomial

The threshold on the degree of the fitted polynomial is 5 and the scatter plots for the results obtained using the corresponding parameter is shown below. The testing RMSE is 0.005068.

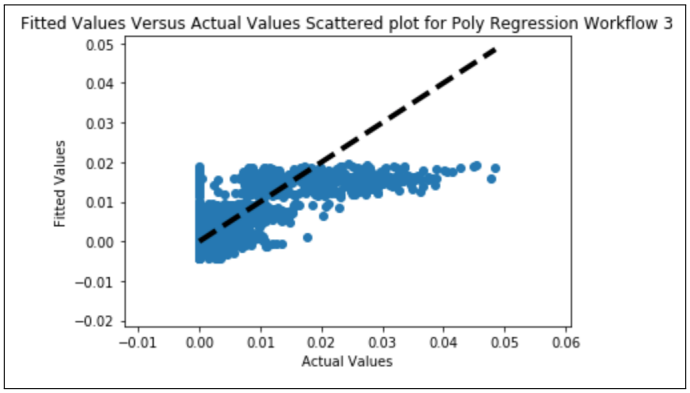


Figure 63: Fitted Values Against Actual Values (Type 3)

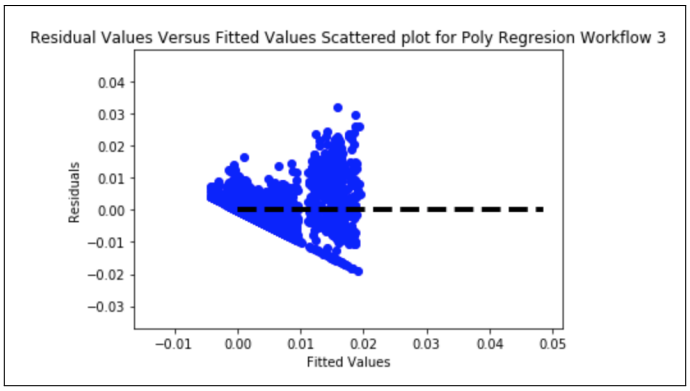


Figure 64: Residuals Against Fitted Values (Type 3)

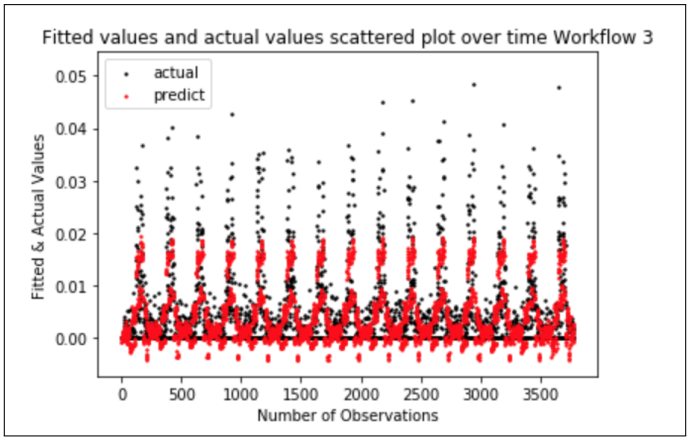


Figure 65: Actual and Predicted Values (Type 3)

3.4.2.5 Workflow Type 4

The training RMSE and testing RMSE are plotted against the degree of the polynomial used.

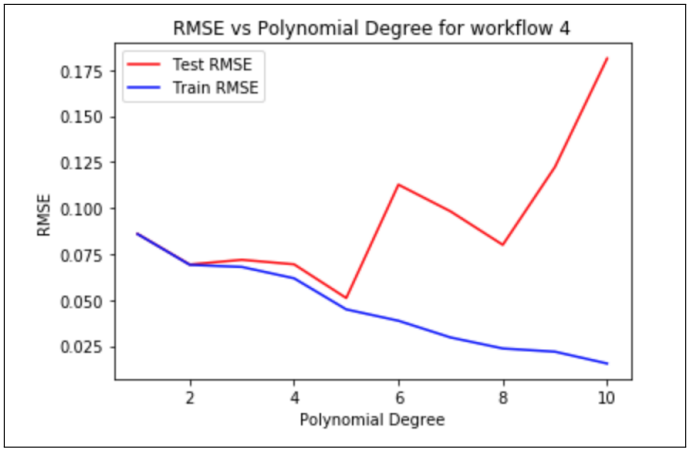


Figure 66: Training and Testing RMSE Against Degree of Polynomial

The threshold is 5 and the scatter plots for the results are shown below. The testing RMSE obtained is 0.05112.

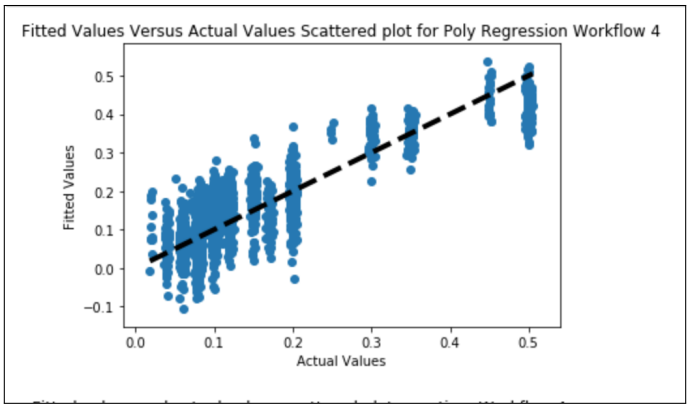


Figure 67: Fitted Values Against Actual Values (Type 4)

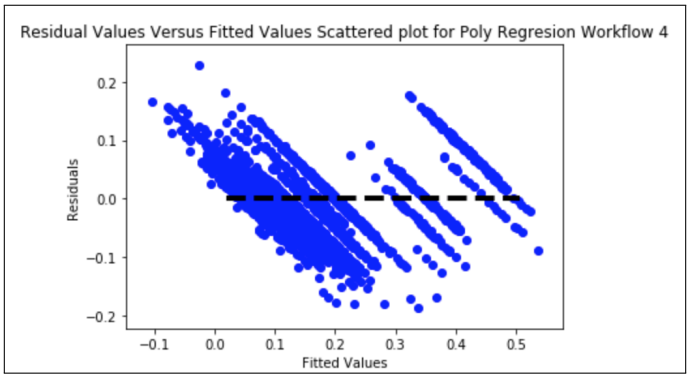


Figure 68: Residuals Against Fitted Values (Type 4)

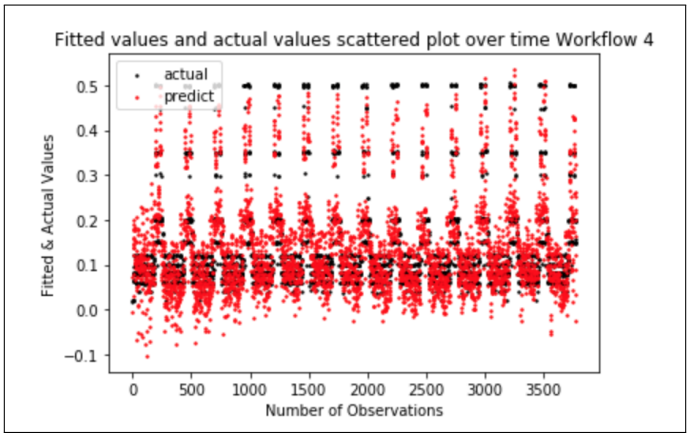


Figure 69: Actual and Predicted Values (Type 4)

3.4.2.6 Analysis

Compared to the results obtained in the previous section, where the variables are applied directly to the input of the linear regression model, using polynomial functions of variables as features significantly improved the accuracy of the predictions.

When polynomial functions of variables are used as input variables, the number of features increases, which might lead to over-fitting. For example, in the 2D plane, a sufficiently high degree polynomial can always perfectly fit all points in the plane. However, the general trend of the data points is not captured as a result. Cross-validation alleviates the problem of over-fitting and hence controls the complexity of the model, as the error will be high in the validation set if over-fitting occurs.

3.5 k-Nearest Neighbors (k-NN) Regression Model

In k-NN regression, the output can be predicted by taking a weighted average of the k-nearest neighbors, weighted by the inverse of their distance. For this model, the Euclidean distance is chosen as the distance metric.

k-NN is carried out for the scalar-encoded and one-hot-encoded input variables.

3.5.1 Scalar Encoding

The best model is determined over varying number of nearest neighbors, k, and the number of leaves. The model with $k = 4$ and number of leaves $= 20$ had the lowest RMSE. The training and testing RMSE are tabulated below.

Training RMSE	0.01684
Testing RMSE	0.03508

Table 14: RMSE for k-NN Using Scalar-Encoded Features

Scatter plots of fitted values against true values, and residuals against fitted values are shown in Figures 70 and 71, respectively.

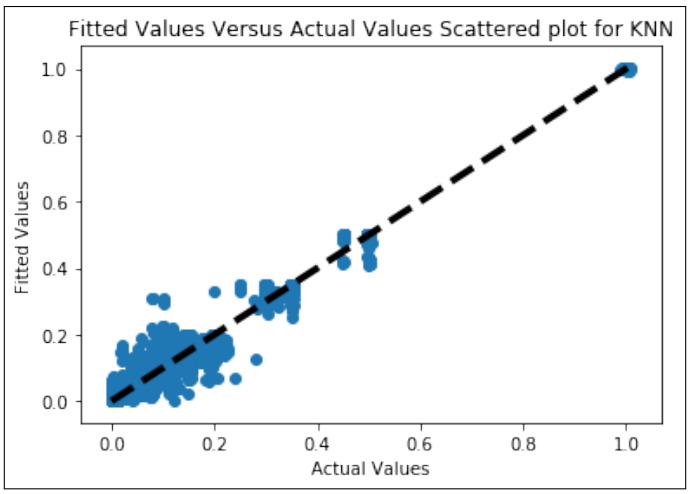


Figure 70: Fitted Values Against Actual Values

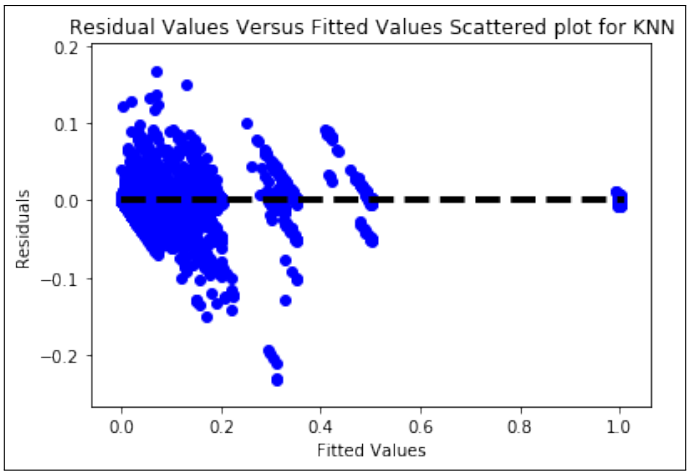


Figure 71: Residuals Against Fitted Values

The actual and predicted values are plotted on the same graph below.

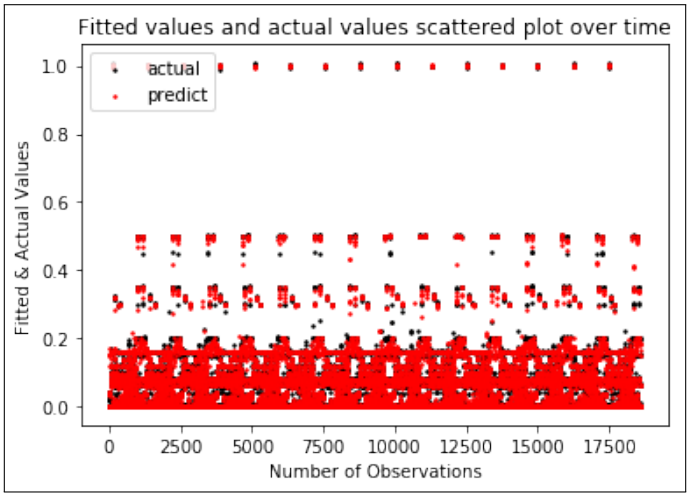


Figure 72: Actual and Predicted Values

3.5.2 One-Hot Encoding

k-NN regression is now carried out using one-hot-encoded feature variables. The model with $k = 4$ and number of leaves = 30 produced the lowest RMSE. The training and testing RMSE are tabulated below.

Training RMSE	0.01073
Testing RMSE	0.01914

Table 15: RMSE for k-NN Using One-Hot-Encoded Features

Scatter plots of fitted values against true values, and residuals against fitted values are shown in Figures 73 and 74, respectively.

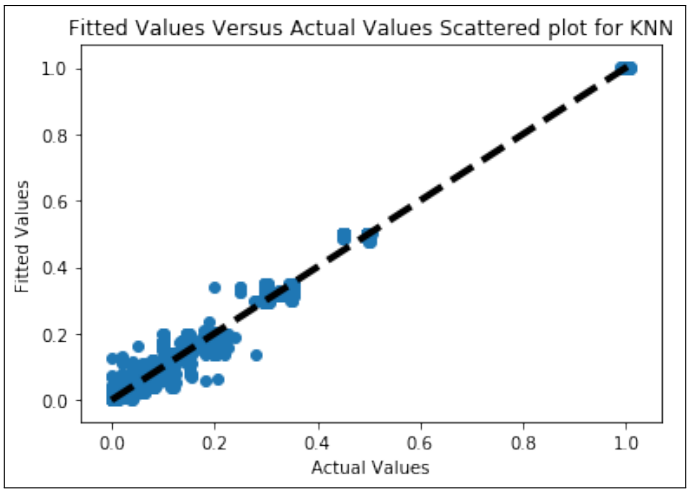


Figure 73: Fitted Values Against Actual Values

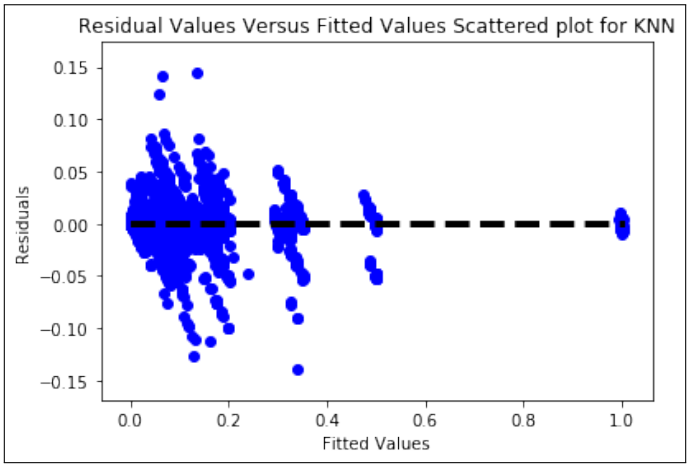


Figure 74: Residuals Against Fitted Values

The actual and predicted values are plotted on the same graph in Figure 75.

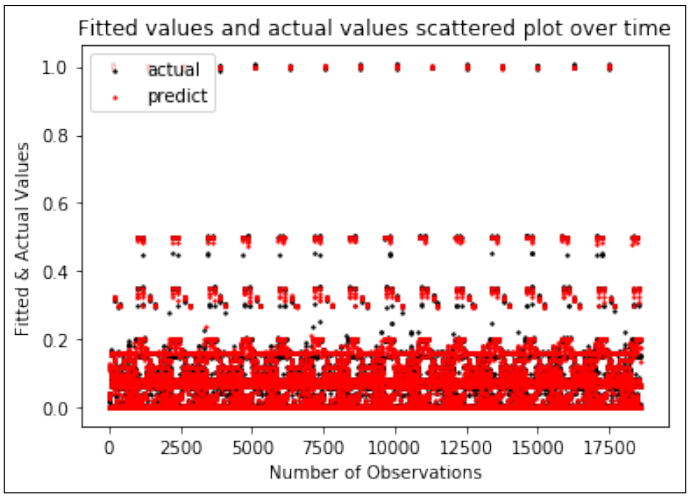


Figure 75: Actual and Predicted Values

The one-hot encoding scheme outperformed the scalar encoding scheme for k-NN regression.

4 Comparison of Regression Models

Table 16 summarizes the testing RMSE for the various regression models explored in this project.

	Testing RMSE
Linear Regression (Scalar)	0.1037
Linear Regression (Combi)	0.08850
Linear Regression (Combi + Ridge Reg)	0.08850
Random Forest Regression (Initial)	0.06062
Random Forest Regression (Best)	0.01293
Neural Network Regression (Best)	0.1027
Linear Regression (Separate) - Average	0.06419
Polynomial Regression (Separate) - Average	0.02165
k-NN Regression (Scalar)	0.03508
k-NN Regression (One-Hot)	0.01914

Table 16: Testing RMSE for Various Regression Models

The optimized random forest regression model produced the lowest testing RMSE. This is expected as random forest is well-known for its robustness. Over-fitting is also less of a problem in this implementation, allowing better fitting on the testing dataset. It is the best at handling categorical features as encoding schemes for the input variables are not required. However, there are certain disadvantages to this method as the prediction is not in a continuous manner and it is unable to predict values beyond the training dataset.

k-NN regression handles sparse features well. This can be seen from the significant improvement in testing RMSE when the feature encoding scheme is changed from scalar to one-hot. One-hot encoding leads to sparse features but the algorithm managed to achieve a better result.

Linear regression is generally unable to handle sparse features. However, one-hot encoding may allow certain input features to be expressed in a more meaningful manner. Hence, there is a trade-off present and with careful selection of encoding schemes, better results can be obtained compared to when using all-scalar-encoded or all-one-hot-encoded feature variables.